12. (Amended) The method of claim 7 wherein said data is selected from a group consisting of data 16, 32, and 64 bits in length.

13. The method of claim 3 wherein said unaligned data is 16 or 32 bits in length, and said first instruction further comprises sign-extension when said unaligned data is in big endian order and said second instruction further comprises sign-extension when said data is in little endian order.

REMARKS

Claims 1-4 and 6-13 are pending. Claim 5 has been canceled without prejudice. Claims 1, 2, 7, 11, and 12 have been amended to correct minor informalities and to more particularly point out and distinctly claim Applicant's invention. New claim 13 has been added. No new matter has been introduced. Applicant believes the claims comply with 35 U.S.C. § 112.

Claims 1-4 and 6-12 stand rejected under 35 U.S.C. § 102(a) as being anticipated by Cho et al. Claims 1-4 and 6-12 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Ray et al. Claims 1-4 and 6-12 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Groves.

Applicant respectfully submits that independent claim 1 is novel and patentable over any of Cho et al., Ray et al., and Groves because, for instance, none of them discloses or suggests providing a first instruction causing loading of a first part of the unaligned data into a first storage location by using a first pointer giving a memory address of a first position; providing a second instruction causing loading of a second part of the unaligned data into a second storage location by using a second pointer giving a memory address of a fourth position; and providing a third instruction causing combining of the first storage location with the second location using a logical operation into a result storage location.

The method as recited in claim 1 employs a simple implementation by providing three separate instructions. The three simple instructions facilitate efficient processing. A more complicated instruction may cause a pipeline to stall, for instance,

when it is hard to write data to a memory continuously. As a result, the control of the process in the such an approach becomes more complicated. On the other hand, the claimed method in such a situation can execute another instruction which does not require memory write. The claimed method avoids the problem by providing three separate instructions in a simple implementation. In contrast, the cited references each employ a more complicated implementation to provide one complex instruction. In addition, the simple implementation reduces complexity, thereby allowing the process to run at higher frequency with the same process technology and reducing the verification cost.

For at least the foregoing reasons, claim 1 and claims 2-4, 6, and 13 depending therefrom are novel and patentable over the cited references.

Applicant respectfully submits that independent claim 7 is novel and patentable over any of Cho et al., Ray et al., and Groves because, for instance, none of them discloses or suggests providing a first instruction causing rotation of data in a first storage location and storing of a first part of the data in a first portion of unaligned plurality of memory locations from a first position to a second position; and providing a second instruction causing rotation of data in a second storage location and storing of a second part of the data in a second portion of unaligned plurality of memory locations from a third position to a fourth position.

The method as recited in claim 1 employs a simple implementation by providing two separate instructions. The two simple instructions facilitate efficient processing. As discussed above, the cited references each employ a more complicated implementation to provide one complex instruction.

For at least the foregoing reasons, claim 7 and claims 8-12 depending therefrom are novel and patentable over the cited references.

CONCLUSION

In view of the foregoing, Applicant believes all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 650-326-2400.

Respectfully submitted,

Chun-Pok Leung
Reg. No. 41,405

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 650-326-2400
Fax: (415) 576-0300
RL:rl
PA 3258403 v1

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION:

Page 5, lines 16-27, please amend the paragraphs as follows:

[Figure] Fig. 11 illustrates a simplified block diagram of the align and sign extension block;

[FIG.] Fig. 12 shows a specific embodiment of a 8 byte Right Rotator 800 of the present invention;

[FIG.] Fig. 13 shows an example of how a 4-byte unaligned word load instruction achieves alignment in different Endianness;

[FIG.] Fig. 14 shows an example of alignment of a 4-byte unaligned store instruction in different Endianness;

[FIG.] Fig. 15 illustrates a simplified block diagram of the Sign selection block 680 of the present invention;

[FIG.] Fig. 16 illustrates a simplified block diagram of the Sign (zero) Extension Block 690 of the present invention.

IN THE CLAIMS:

Please cancel claim 5 without prejudice; amend claims 1, 2, 7, 11, and 12; and add new claim 13 as follows.

1. (Amended) A method for loading unaligned data stored in a plurality of memory locations, comprising:

providing a first instruction causing loading of a first part of said unaligned data into a first storage location by using a first pointer giving a memory address of a first position;

rotating and masking said first part of said unaligned data in said first storage location from a first position to a second position;

providing a second instruction causing loading of a second part of said unaligned data into a second storage location by using a second pointer giving a memory address of a fourth position;

rotating and masking said second part of said unaligned data in said second storage location from a third position to a **[forth]** fourth position; and

providing a third instruction causing combining of said first storage location with said second location using a logical operation into a result storage location.

2. (Amended) The method of claim 1 wherein said first pointer **[storage location]** is a first register, said **[second]** first storage location is a second register, said second pointer is a third register, and said second storage location is a fourth register, and said result storage location is a result register.

5. CANCELED.

7. (Amended) A method for storing data into an unaligned plurality of memory locations, comprising:

providing a first instruction causing rotation of **[rotating a first part of said]** data in a first storage location and storing of a first part of said data in a first portion of unaligned plurality of memory locations from a first position to a second position;

[storing said data located in second position in said unaligned plurality of memory locations at an address given by] having a first pointer giving an address of a first position;

providing a second instruction causing rotation of **[rotating a second part of said]** data in a second storage location and storing of a second part of said data in a second portion of unaligned plurality of memory locations from a third position to a **[forth]** fourth position; and

[storing said data located in forth position in said unaligned plurality of memory locations at an address given by] having a second pointer giving an address of a fourth position.

11. (Amended) The method of claim 7 wherein said first storage location is a first **[64-bit]** register, said **[second storage location is a 64-bit second**

register, and said result storage location is a 64-bit result] first pointer is a second register, and said second pointer is a third register.

12. (Amended) The method of claim 7 wherein said data is selected from a group consisting of data **[8,]** 16, 32, and 64 bits in length.

--13. The method of claim 3 wherein said unaligned data is 16 or 32 bits in length, and said first instruction further comprises sign-extension when said unaligned data is in big endian order and said second instruction further comprises sign-extension when said data is in little endian order.--